Ojo Podo

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 2

1.1 Purpose 2

1.2 Scope 2

1.3 Definitions, Acronyms, and Abbreviations 2

1.4 References 2

1.5 Overview 2

2. General Use-Case-Modeling Guidelines 2

3. How to Describe a Use Case 2

# Introduction

## Purpose

Tujuan dari pedoman ini adalah untuk memastikan konsistensi pemodelan use case. Dokumen ini menyediakan arahan bagaimana mendokumentasikan sebuah use case dan membantu masalah-masalah yang dihadapi berkaitan dengan hal ini kepada penentu kebutuhan dan sistem analis.

## Scope

Pedoman ini digunakan, atau disesuaikan, untuk memenuhi kebutuhan proyek.

## Definitions, Acronyms, and Abbreviations

## References

## Overview

Dokumen ini terbagi atas dua bagian. Bagian pertama menjelaskan cara kami memodelkan Use-case, dan bagian kedua menyediakan pedoman untuk isi dari model Use-case dan untuk penamaan elemen-elemen di dalam model.

# General Use-Case-Modeling Guidelines

## Bentuk Umum

Use Case akan ditulis sesuai dengan template yang disediakan oleh Rational Unified Process, dengan bentuk tertentu dan modifikasi tampilan untuk menyesuaikan standar dokumentasi proyek aplikatif.

## Kegunaan hubungan <<Communicates>>

Asosiasi antara sebuah Actor dan sebuah Use Case disebut dengan sebuah hubungan Communicates. Direkomendasikan bahwa asosiasi ini dibuat uni-directional. Dengan menggunakan strategi pemodelan ini kita dapat membedakan antara:

1. Aktor Aktif

Sebuah aktor dikatakan aktif pada sebuah pasangan Aktor-Use Case ketika aktor menginisiasi (atau memancing) eksekusi dari Use Case. Panah pada hubungan Communicates ini mengarah ke Use Case.

1. Aktor Pasif

Sebuah aktor dikatakan pasif pada sebuah pasangan Aktor-Use Case ketika Use Case yang bertugas menginisiasi komunikasi. Aktor pasif biasanya berupa sebuah sistem eksternal atau device yang dibutuhkan untuk berkomunikasi dengan sistem kita. Panah pada hubungan Communicates ini mengarah ke Aktor.

Rekomendasi ini dibuat karena pengartian mengenai aktor aktif dan pasif menambah pemahaman bagi pembaca model Use Case.

## Kegunaan hubungan <<Include >> dan <<Extend>>

Dalam instansi pertama, direkomendasikan kepada Anda untuk menghindari penggunaan kedua hubungan ini. Rekomendasi ini dibuat karena penyalahgunaannya dapat berpotensi mengacaukan dan membingungkan daripada untuk membantu menyederhanakan model Use Case. Cara terbaik adalah menghindari penggunaan bentuk dekomposisi ini pada inisiasi, dan mempertimbangkan untuk menggunakannya pada bagian terakhir proses. Hubungan ini dapat digunakan untuk:

1. Menemukan perilaku yang digunakan dalam dua atau lebih use case.
2. Menemukan perilaku dari use case dasar yang tidak diperlukan untuk pemahaman terhadap tujuan utama dari use case. Hanya hasil dari perilaku itu yang penting.
3. Menampilkan suatu set perilaku yang satu atau beberapa diantaranya dapat dimasukkan dalam sebuah titik extensi di dalam sebuah use-case dasar.

Tetapi hubungan-hubungan ini hanya digunakan apabila dapat membantu untuk memudahkan dan mengatur model use-case.

### Kegunaan hubungan <<Include>>

Hubungan include menggambarkan suatu perilaku yang dimasukkan ke dalam sebuah instansi use-case yang mengeksekusi use-case dasar. Include adalah sebuah mekanisme yang mirip dengan sub-routine, dan paling banyak digunakan untuk menemukan perilaku umum.

### Kegunaan hubungan <<Extend>>

Hubungan extend lebih sulit dimanfaatkan, biasanya karena use-case ekstensi tidak dikenal oleh use-case dasar. Pada beberapa tempat hubungan ini dapat berguna pada banyak sistem bisnis. Harus tetap dipahami bahwa selalu ada pengecualian pada sebuah aturan, dan mekanisme extend ini dapat berguna pada kondisi-kondisi tertentu.

## Kegunaan Auto-Generalization

Umumnya, Auto-Generalization dapat digunakan untuk mendefinisikan peran-peran berbeda yang dimainkan oleh pengguna dari sistem yang akan dikembangkan. Ini sangat berfungsi untuk aplikasi-aplikasi dengan kategori pengguna yang berbeda-beda. Dengan cara ini, hanya fungsi yang relevan yang diberikan kepada tiap kategori pengguna, dan kita dapat mengontrol hak akses berdasarkan pengelompokan tersebut.

## Kegunaan diagram interaksi

Dalam beberapa kasus, sangat bermanfaat untuk memasukkan sebuah diagram Interaksi untuk mengilustrasikan alur kejadian “tingkat tinggi” sebuah use-case. Direkomendasikan bagi Anda untuk menggambar diagram sekuensial untuk ini dalam Rational Rose. Masukkan hanya komunikasi antara aktor dan objek pembatas (mencakup pesan input dan output) dan perlakukan sistem sebagai sebuah black box. Gunakan objek boundary dengan nama yang logik sebagaimana terdefinisi di dalam alur kejadian use case, tanpa menempatkan mereka dalam kelas-kelas terlebih dahulu.

Tidak terlalu diperlukan untuk membuat diagram interaksi pada tiap use-case. Diagram interaksi bersifat opsional.

## Kegunaan diagram aktivitas

Apabila diagram aktivitas diperlukan dalam membantu mendefinisikan, mengklarifikasi, dan melengkapi alur kejadian dalam use-case, maka direkomendasikan untuk dimodelkan dalam Rational Rose. Diagram aktivitas dapat sangat berguna untuk use-case yang rumit. Diagram aktivitas menampilkan sebuah decision tree dari alur-alur di dalam use-case.

Tidak terlalu diperlukan untuk membuat diagram aktivitas pada tiap use-case. Diagram aktivitas bersifat opsional.

# How to Describe a Use Case

## Pedoman Aktor

### Tiap use-case melibatkan sedikitnya satu aktor

Apakah tiap use-case melibatkan sedikitnya satu aktor? Jika tidak, berarti ada sesuatu yang salah. Sebuah use-case yang tidak berinteraksi dengan satu pun aktor adalah tidak berarti. Anda dapat memilih untuk menghapus use-case tersebut atau mencari aktor yang berhubungan.

Pada beberapa kasus, lebih dari satu aktor dapat memainkan satu bagian dalam interaksi use-case. Pastikan memeriksa bahwa penggunaan penggunaan banyak aktor dalam satu use-case sudah valid.

### Nama aktor intuitif dan deskriptif

Apakah nama aktor sudah intuitif dan deskriptif? Dapatkah pihak pengguna dan klien memahami nama-nama aktor? Sangat penting untuk menyesuaikan nama aktor dengan peran yang dipegang. Jika tidak, ganti namanya.

Anda harus merujuk ke Use Case Model untuk memastikan bahwa Anda menggunakan nama aktor yang benar pada tiap use-case.

### Konsistensi penggunaan nama aktor

Spesifikasi use-case akan ditulis dengan nama aktor secara konsisten. Pastikan penamaan aktor jelas dan tidak ambigu.

Jangan merujuk langsung ke “Aktor”; melainkan gunakanlah nama tertentu untuk mendefinisikan atau mengidentifikasikan secara unik sebuah aktor. Nama aktor dapat dikembangkan dari peran yang dimainkan pada interaksi sistem.

## Nama Use-case

Nama use-case haruslah unik, intuitif, dapat menjelaskan sehingga jelas dan tidak ambigu mendefinisikan hasil yang didapatkan dari use-case.

Pemeriksaan yang baik pada nama use-case adalah untuk mengetahui bahwa klien, perwakilan bisnis, analis, dan developer memahami nama-nama dan deskripsi dari use case.

Tiap nama use-case akan menjelaskan perilaku yang didukung use-case. Nama akan berupa kombinasi dari aksi yang dijalankan dan elemen kunci yang dijalankan. Paling sering, nama use case merupakan kombinasi kata kerja dan kata benda. Use-case harus dinamakan dari perspektif aktor yang menggerakkan use-case.

## Penjelasan singkat use-case

### Minimal satu paragraf

Use case akan berisi sebuah penjelasan singkat. Penjelasan memiliki panjang minimal satu paragraf dan tidak lebih dari tiga paragraf. Penjelasan mencakup pembahasan mengenai tujuan utama, proposisi nilai, dan konsep dari use case.

### Dilengkapi contoh (opsional)

Apabila sangat berguna, dapat ditambahkan sebuah contoh untuk menambah pemahaman mengenai konteks use-case.